AUTOMATIC DETECTION OF ARROW ANNOTATION OVERLAYS IN BIOMEDICAL IMAGES

Beibei Cheng[a], R. Joe Stanley[a], Soumya De[a], Sameer Antani[b], George R. Thoma[b]

[a]Department of Electrical and Computer Engineering

Missouri University of Science and Technology

Rolla, MO 65409-0040 USA

[b]Lister Hill National Center for Biomedical Communications

National Library of Medicine, National Institutes of Health, DHHS

Bethesda, MD 20894 USA

ABSTRACT

Images in biomedical articles are often referenced for clinical decision support, educational purposes, and medical research. Authors-marked annotations such as text labels and symbols overlaid on these images are used to highlight regions of interest which are then referenced in the caption text or figure citations in the articles. Detecting and recognizing such symbols is valuable for improving biomedical information retrieval. In this research, image processing and computational intelligence methods are integrated for object segmentation and discrimination and applied to the problem of detecting arrows on these images. Evolving Artificial Neural Networks (EANNs) and Evolving Artificial Neural Network Ensembles (EANNEs) computational intelligence-based algorithms are developed to recognize overlays, specifically arrows, in medical images. For these discrimination techniques, EANNs use particle swarm optimization and genetic algorithm for artificial neural network (ANN) training, and EANNEs utilize the number of ANNs generated in an ensemble and negative correlation learning for neural network training based on averaging and Linear Vector Quantization (LVQ) winner-take-all approaches. Experiments performed on medical images from the imageCLEFmed'08 data set, including 395 images with one or more arrows and 288 images with no arrows, yielded area under the receiver operating characteristic curve and precision/recall results as high as 0.988 and 0.928/0.973, respectively, using the EANNEs method with the winner-take-all approach.

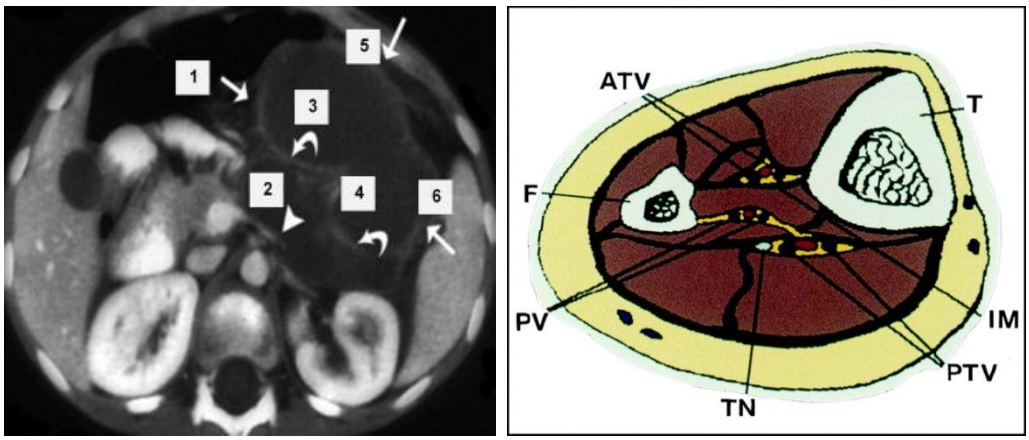
1. INTRODUCTION

Authors of biomedical publications use images to illustrate medical concepts and highlight special cases. These images often convey essential information and can be very valuable for improved clinical decision support (CDS) and education. Biomedical information retrieval has, so far, been largely text-based and limited mostly to bibliographic information. To be of greater value, it is desirable to retrieve images from biomedical publications. However, they need to be first annotated with respect to their usefulness for CDS to help determine relevance to a clinical query or to queries for special cases important in educational settings (Demner-Fushman, 2007, 2008, 2009).

Image retrieval can be achieved using the following methods: (i) traditional text-based approaches that index figure captions, (ii) image retrieval approaches that index the visual content of the images, and (iii) an intelligent combination of the above. To enhance text-based retrieval, content-based image retrieval (CBIR) has been explored to retrieve information from images in the biomedical field (Demner-Fushman, 2007). However, the approaches have not taken advantage of specific image regions of interest (ROIs) highlighted by the author using overlaid symbols, such as arrows and other text labels, and identifying them in the caption text. Further, it has been shown that whole image retrieval without attention to specific regions of interest marked by annotations, such as arrows (Figure 1), is not as promising as retrieval of text, primarily due to "semantic gap" introduced by less relevant image regions (Deserno, 2009). It is commonly understood in the field that low level features such as color, texture, and shape used in CBIR are insufficient to represent medical concepts or meaningful diagnostic information in the images effectively unless they can be applied to the key image regions such as those identified by the author, as in the case of images from biomedical articles. To improve the relevance quality of conventional retrieval approaches, we have proposed an approach using hybrid (text and image) features (Antani, 2008; You, 2009, 2010). Information retrieval techniques are used to identify key textual features in the title, abstract, figure caption, and figure citation ("mention") in the article. Structured vocabularies, such as the National Library of Medicine's Unified Medical Language System (UMLS®) are used as well to identify the biomedical concepts in these (Demner-Fushman, 2009; You, 2009). Unlike conventional CBIR schemes that extract features from the entire image, our approach uses a combination of features: those computed from specific image region of interests (ROIs) in addition to the ones obtained from the entire image. The ROIs are detected by localizing and recognizing image annotations

such as arrows overlaid by authors. Annotations and ROIs in retrieved images can be identified by the annotation recognizer and then used to re-rank the results of the recognizer.

There are some techniques that have been implemented to find arrows in previous research. Sparce pixel vectorization has been explored to detect arrowheads (Dov, 1999). In addition, arrow sign identification has been investigated for robot navigation using a camera-based method (Park, 2008). Compared to the existing approaches, the arrow symbols seen in the medical images experimental data set used in this research have a more complex shape. As shown in Figure 1(a), arrows in these medical images do not necessarily have to be straight (see arrow 3, arrow 4 in Figure 1(a)) and the shape of the arrows can be significantly different (see arrow 2 in Figure 1(a)) as well. Furthermore, the example image in Figure 1(b) shows objects such as characters and symbols which can be of similar size to arrows, providing potential false arrow detections. Therefore, a general and robust arrow detection algorithm is needed for discrimination from other medical image artifacts.



(a) (b)

Figure 1. Medical image with arrows. (a) Arrow example (Adapted from (Caskey, 1999)). (b) Non-arrow example (Adapted from (Fraser, 1999)).

This work extends previous research for a computational intelligence-based approach for medical image symbol (arrow) analysis (Cheng, 2010). In this research, a data set of 683 medical images annotated by modality (radiological, photo, etc.), was selected from the

imageCLEFmed'08 (http://www.imageclef.org) data set, including 395 images with one or more arrows and 288 images with no arrows (Müller, 2010). An overview of the arrow detection analysis process for medical images is shown in Figure 2. Since arrow, text and symbol objects are white or black, they can be segmented using image analysis techniques. After generating the binary image containing only text-like and symbol-like objects, feature sets are used as input to classifiers so that we can discriminate the arrows from noise and other types of medical symbols (Cheng, 2010). The various steps in the flowchart presented in Figure 2, are explained in Section 2. Section 3 gives the experimental results, Section 4 provides the discussion, and Section 5 presents conclusions and future work.

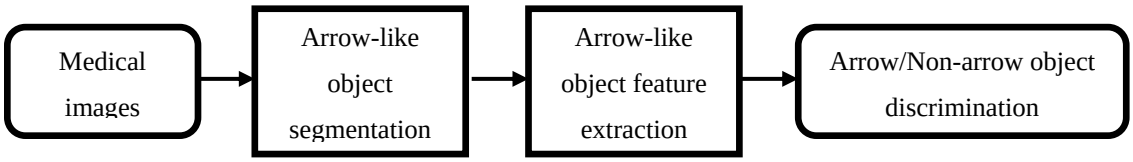


Figure 2. Overview of arrow detection process.

2. ARROW DETECTION PROCESS

As previously stated, medical images can contain arrow, text, and symbol objects. From these images, arrows typically are white or black objects as distinguished color from the background. These arrow objects can be segmented based on grayscale thresholding and edge detection, which is the basis for the segmentation algorithm presented. Thresholding and edge detection provide complementary information for finding arrow-like objects within the image with similar gray levels and potentially varying contrast with the surrounding background. Arrow-like object analysis and pruning are performed using extracted features with computational intelligence techniques. In the following sections of the paper, the algorithmic details are given for the methods used in the different blocks of arrow detection process flowchart from Figure 2.

2.1 OBJECT SEGMENTATION

From the medical images, the initial step is to segment arrow-like objects using a combination of thresholding and edge detection techniques. The object segmentation algorithm consists of the following steps:

1) Convert RGB images into luminance grayscale images.
2) Use Otsu's method (Otsu, 1979) to generate a preliminary object mask for arrow-like objects.
3) Remove objects that are considered small (pixel number of the object area is less than 60) from the preliminary mask in Step 2.
4) Generate an edge image of arrow-like objects using a gray drop method, extending the algorithm developed in (Cheng, 2011). If the absolute gray value of the center pixel (C) minus the gray value of NW, N, NE, W, E, SW, S, SE (see Figure 3) is greater than the gray drop, determined experimentally as 30, this pixel will be marked in the edge image. Figure 3 shows the edge detection operator mask.
5) Compute the bounding boxes of the objects in the masks from Steps 3 and 4.
6) Compare bounding box sizes for corresponding objects from the masks in Steps 3 and 4 and retain the objects with the same bounding box size. Let R_G denote the mask image determined from the grayscale image.
7) Repeat Steps 1-6 for inverted grayscale images since arrow, text and symbol objects may also be black. Let R_I denote the mask image determined from the inverted grayscale image.

8) Compute the final arrow-object mask as the OR image of R_G and R_I, denoted as $R_{G+I} = R_G + R_I$.

Figure 4 presents an image example of the image processing steps for the original image to generate the binary mask for feature calculations. Note that Figure 4 (i) and (k) are empty images because there are no arrow-like objects resulting from these steps in the image process steps to find arrows.

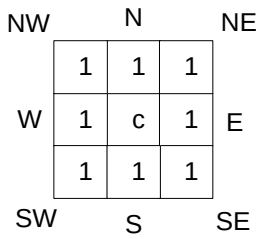


Figure 3. Edge detection operator mask.

2.2. FEATURE EXTRACTION

After completing the image processing steps for arrow-like object segmentation, features are extracted from each object. In (Park, 2008), features including extent and solidity were selected for arrow discrimination and line segment features were utilized to estimate the orientation of arrow objects (You, 2010). For the (You, 2010) study, arrow orientation, not detection, was explored. A typical arrow has a head region with varying stem types. Variations of arrow heads and arrow stems can be observed in Figure 1 (a). In order to address the complexity of the size and shape variations of the arrows in the medical images for the experimental data set, features in three categories are examined, including region property features, shape features and correlation-based features. To this end, multiple features are computed for each object in the mask *RG+I* (see Step 8 in Section 2.1) to facilitate arrow/non-arrow discrimination by using the object mask image computed as shown in section 2.1. The feature set descriptions are given as follows.

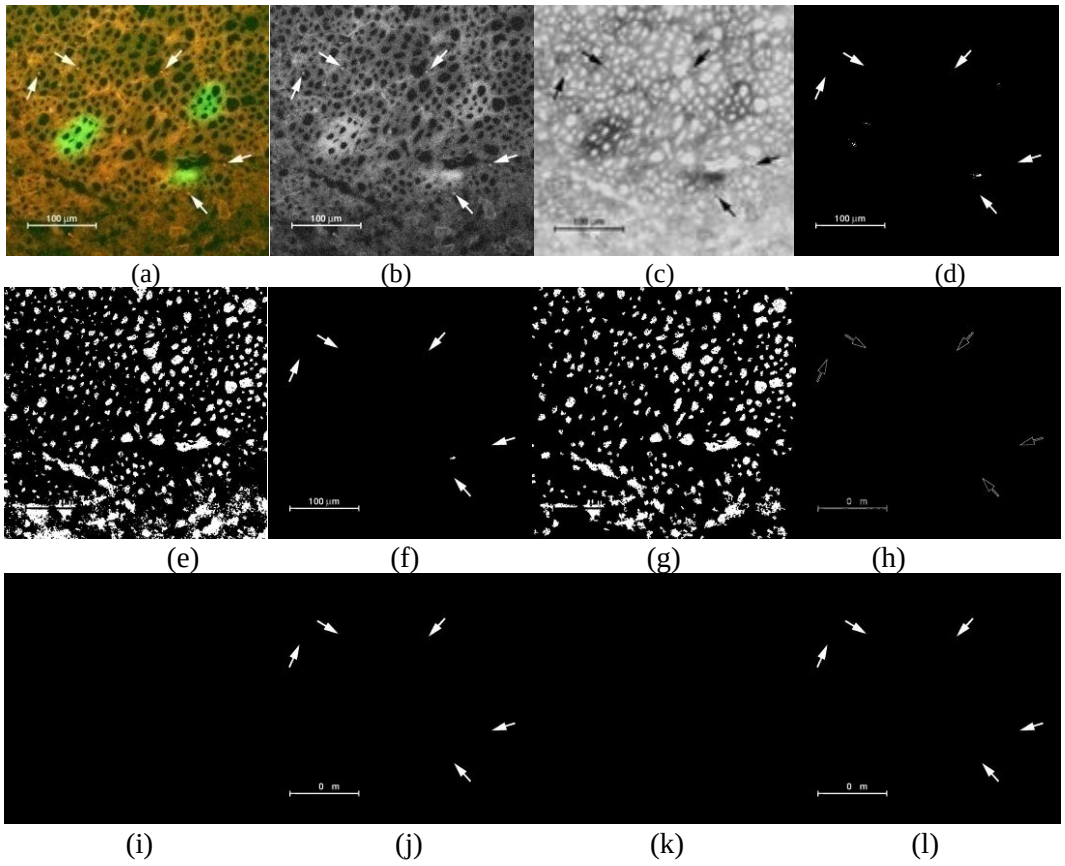


Figure 4. Object segmentation example. (a) Original image (Adapted from (Li, 2003)). (b) Grayscale image. (c) Inverted grayscale image. (d) Gray threshold image for (b). (e) Inverted gray threshold image for (c). (f) Gray threshold image after noise removal. (g) Inverted gray threshold image after noise removal. (h) Gray edge image for (b). (i) Inverted gray edge image for (c). (j) Gray image comparing (f) to (h) with the bounding box size. (k) Inverted gray image comparing (g) to (i) with the bounding box size. (l) Final OR-image of (j) and (k).

2.2.1 Region Property Features. The first set of features is based on the region properties and is computed using the Matlab® function regionprops (Hanselman, 2004). The regionprops features represent the visualization of the objects based on:

- ***MajorAxisLength***: length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region.
- ***MinorAxislength***: length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region.
- ***Axis Ratio***: ratio of *MajorAxisLength* to *MinorAxislength*.

- ***Normalized area***: area of the region divided by the whole image.
- ***Solidity:*** area of the region divided by the convex hull area.
- ***EulerNumber:*** equal to the number of objects in the region minus the number of holes in those objects.
- ***EquiDiam***: the diameter of a circle with the same area as the region.
- ***Extent:*** ratio of area to bounding box area.

2.2.2 Shape Features. The second set of features computed for the arrow-like objects are shape features. These features include:

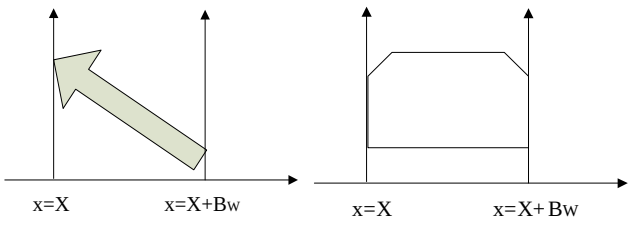
- ***AvgSkelDist:*** average width of object. It can be expressed in the following equation:

$$AvgSkelDist = \sum_{s=1}^{S} D_s / S \quad (1)$$

The skeleton of the arrow-like object was determined using the morphological skeleton algorithm (Serra, 1982). S is the total number of the pixels inside the skeleton object. D_s is the distance from pixel inside the skeleton object to the nearest pixel in the boundary of the object.

- ***MinPixelNo:*** the minimum number of intersection areas for the object and the two lines as shown in Figure 5. The function of these two lines are $x=X$ and $x=X+B_W$ (X is the left column of the bounding box; B_W is the width of bounding box), which is shown in Equation 2. The value of *MinPixelNo* for arrow (Figure 5(a)) is usually smaller than the value for noise (Figure 5(b)) due to the shape of the arrowhead.

$$MinPixelNo = \min(Area_{and(x-X,Object)}, Area_{and(x-X+B_W,Object)}) \quad (2)$$



(a) (b)

Figure 5. MinPixelNo feature. (a) Arrow. (b) Noise.

2.2.3 Correlation-based Features. The third set of features is based on computing shape profiles of binary arrow-like objects and correlating those profiles with basic functions. A one dimensional shape profile of each arrow-like object is found as follows. The bounding box of the arrow-like object is determined. Let $\boldsymbol{B_W}$ and $\boldsymbol{B_H}$ denote the bounding box height and width, respectively. The profile at each sample, $\boldsymbol{L(i)}$, is defined as Equation 3.

$$L(i) = \sum_{j=1}^{B_W} R_{G+I}\,(i,j) \tag{3}$$

for $i = 1, \ldots, B_H$. An example of the samples used for profile calculation is given in Figure 6. Let $L = \{L(1), L(2), \ldots, L(B_H)\}$ be the sequence of profile values. Correlation-based features are extracted by correlating the shape profile of the arrow-like object with weighted density distribution (WDD) functions (Piper, 1989), shown in Figure 7. Let W_1 denote the WDD function in Figure 7(a), W_2 denote the WDD function in Figure 7(b), and so on. In previous research, WDD functions have been explored in previous research for: 1) landmine discrimination based on 1-dimensional profiles of metal detector signals (Stanley, 2002) and 2) dermatology skin lesion discrimination based on a 1-dimensional histogram representation of skin lesions (Stanley, 2008). In both previous research applications, WDD functions provided shape-related information in the determination of correlation-based, size-variant, spatially distributed features from 1-dimensional profiles for object discrimination. In this research, the WDD functions have been applied to 1-dimensional projections profiles of arrow-like objects to extract shape information such as symmetry of the objects for object discrimination. These WDD functions account for the degree of change in the spatial distribution encapsulated in a 1-dimensional profile as well as the symmetry of those values for arrow discrimination. The twelve correlation-based features are computed as follows.

Six WDD features $(f_1, \ldots, f_6)$ are computed using the profile L according to the following expression:

$$f_k = \sum_{i=1}^{B_H} L(i) W_k(i) \tag{4}$$

for $k = 1,2, \ldots,6$. Six additional features $(f_7, \ldots, f_{12})$ are computed by correlating the six WDD functions with the sequence of absolute differences between samples value as follows:

$$f_k = \sum_{i=1}^{B_H} |L(i) - L(i-1)| W_k(i) \quad (5)$$

for $k = 1, \ldots, 6$ and $L(0) = 0$.

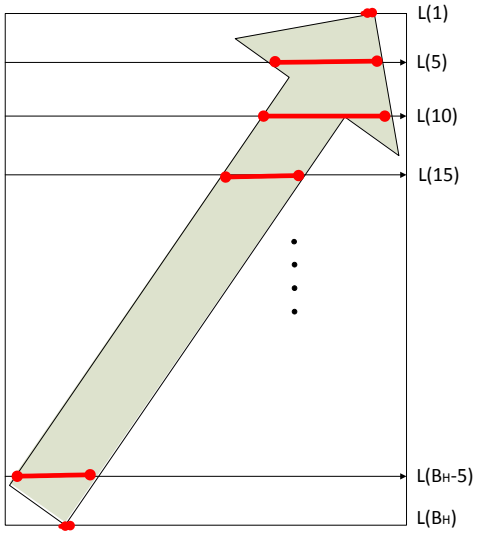


Figure 6. Samples for generating WDD features. (B_H is the height of bounding box)

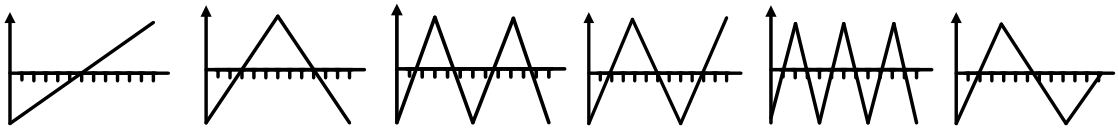


Figure 7. The WDD functions used to compute arrow features (Adapted from (Stanley, 2008)).

2.3 CLASSIFIER ALGORITHMS

Using the features computed for each arrow-like object within the object masks for all images in the experimental data set, Evolving Artificial Neural Networks (EANNs) and Evolving Artificial Neural Network Ensembles (EANNEs) are examined for arrow/non-arrow discrimination. A brief overview for each algorithm is presented in this section, while the implementation details are presented in the Appendix.

EANNs refer to a class of artificial neural networks (ANNs) in which evolution is another fundamental form of adaptation, in addition to learning (Yao, 1993). Evolutionary algorithms (EAs) are used to perform various tasks, such as connection weight training, architecture design, learning rule adaptation, input feature selection, connection weight initialization and rule extraction from ANNs. In our implementation, the Particle Swarm Optimization (PSO) (Kennedy, 1995) method and the Genetic Algorithm (GA) (Holland, 1975) method are explored for connection weight training. Both the PSO and GA use the same scheme for candidates' representation, where each candidate is a D-dimensional vector with each element of the vector representing a connection weight and D being the total number of weights. The population is initialized in the sense that each element in a vector is randomly setting a value from -0.1 to +0.1. The fitness values for PSO and GA are set to be the root mean square error (RMSE) given the connection weights. The RMSE is calculated based on the difference between the grand truth and the actual ANN's output. The algorithms for training the connection weights for ANNs in PSO and GA are explained in detail in the Appendix section.

Evolving Artificial Neural network ensembles (EANNEs) provide a method for utilizing and combining the outputs of several networks (Yao, 2008). Each ANN has the same inputs and generates its own outputs for decision making. The ensemble method is based on the premise that a population contains at least as much information as any single individual. There are two components for ensemble design: 1) a method of training the networks to encourage the diversity of behaviors and 2) a mechanism to decide the final output based on the outputs from the individual networks. For the first component, a cooperative ensemble learning system (CELS) is used for training individual networks. CELS is used to create negatively correlated ANNs using a correlation penalty term in the error function of each individual network so that the mutual information among the networks in the ensemble can be minimized based on the Liu and Yao approach (Liu, 1999). For the second components, since the outputs of the ANNs are floating-point numbers, averaging and winner-taking-all for combining/aggregating the outputs were examined. The algorithms for each component are shown in the Appendix.

3. EXPERIMENTS PERFORMED

The experimental data set consisted of 683 medical image annotated by modality (radiological, photo, etc.) selected from the imageCLEF08 data set (Müller, 2010), including 395 images with one or more arrows and 288 images with no arrows. These images were manually assigned as arrow/no-arrow images for creating the ground-truth database. The object segmentation for arrow-like object segmentation, feature extraction from those objects, and arrow/no arrow discrimination algorithms presented in Section 2 were applied to the image set. Using the object segmentation algorithm from Section 2.1, 724 arrow objects and 1450 text/noise objects were segmented from those images. The 22 input features computed from each arrow-like object include 8 region property features, 2 shape features, and 12 correlation-based shape profile features. Standard backpropagation ANNs, EANNs, and EANNEs algorithms with variations were investigated for arrow/non-arrow discrimination. In order for an object to be scored correctly as an arrow object, the object had to be labeled by the discrimination algorithm as an arrow object, the object had to be completely segmented, and the object had to be ground truthed as an arrow object.

Seven different feature combinations are investigated as inputs to the multilayer perceptrons(MLP) neural network architectures, with the neural network architectures summarized in the following cases: 1) 9x5x1 consisting of an input layer of 8 region property features and a bias with linear neurons, a hidden layer of 5 neurons with sigmoid transfer functions, and an output layer of one output with a linear neuron; 2) 3x5x1 consisting of an input layer of 2 shape features and a bias with linear neurons; 3) 13x5x1 consisting of an input layer of 12 correlation-based features and a bias with linear neurons; 4) 11x5x1 consisting an input layer of combined 8 region property features with 2 shape features and a bias with linear neurons; 5) 21x5x1 consisting of an input layer of combined 8 region property features with 12 correlation-based features and a bias with linear neurons; 6) 15x5x1 consisting of an input layer of combined 2 region property features with 12 shape features and a bias with linear neurons; 7) 23x5x1 consisting of an input layer of combining all three feature groups and a bias with linear neurons. These architectures are summarized in Table 1. A ten-fold cross validation methodology is used for generating training/test sets for each neural network's architecture (Kohavi, 1995). The same training/test sets from the cross-validation process are applied to all feature combinations and classification algorithms presented. Classification results are based on averaging the area under Receiver Operating Characteristic (ROC) curves (Fogarty, 2005) generated for each of the ten test sets. The area under the ROC curve was given as the

evaluation measure because it does not require selecting a decision boundary or threshold to determine detection accuracy. In addition, experimental results are reported using precision and recall (Bar-Ilan, 1998). Here, precision is defined as the number of arrow objects correctly called arrow objects (true positive classifications) divided by the total number of objects called arrow objects, and recall is defined as the true positive classifications divided by the sum of the true positive classifications and the number of objects which were not classified as arrow objects but should have been, i.e. the total number of arrows in the evaluation set of images.

Table 1. Seven cases with their feature combinations and NN architectures.

Case No.	Feature Combination	NN Architecture
1	Region property features	9x5x1
2	Shape features	3x5x1
3	Correlation-based features	13x5x1
4	Region property& Shape features	11x5x1
5	Region property & Correlation-based features	21x5x1
6	Shape & Correlation-based features	15x5x1
7	Region property& Shape & Correlation-based features	23x5x1

Figure 8 presents the ROC curve results for a representative test set for case 1 for the different classifiers with M=5 and N=75 (except for the backpropagation ANN algorithm). Table 2 shows the area under ROC curve results and precision and recall (given in parentheses) for the seven different input features combinations (Case 1 to Case 7) for the EANN, EANNE, and standard backpropagation ANN classifiers investigated. For the EANN and EANNE classifiers, Table 2 includes the results for different population size (M) and the maximum number of generations (N), and the standard online backpropagation ANNs were trained over 2000 epochs. Area under the ROC curve and precision and recall are presented based on averaging the results over the ten test sets from the ten-fold cross validation process. Precision and recall are presented based on specifying precision as a constant of at least 90% (based on the ROC curve) and computing the corresponding recall.

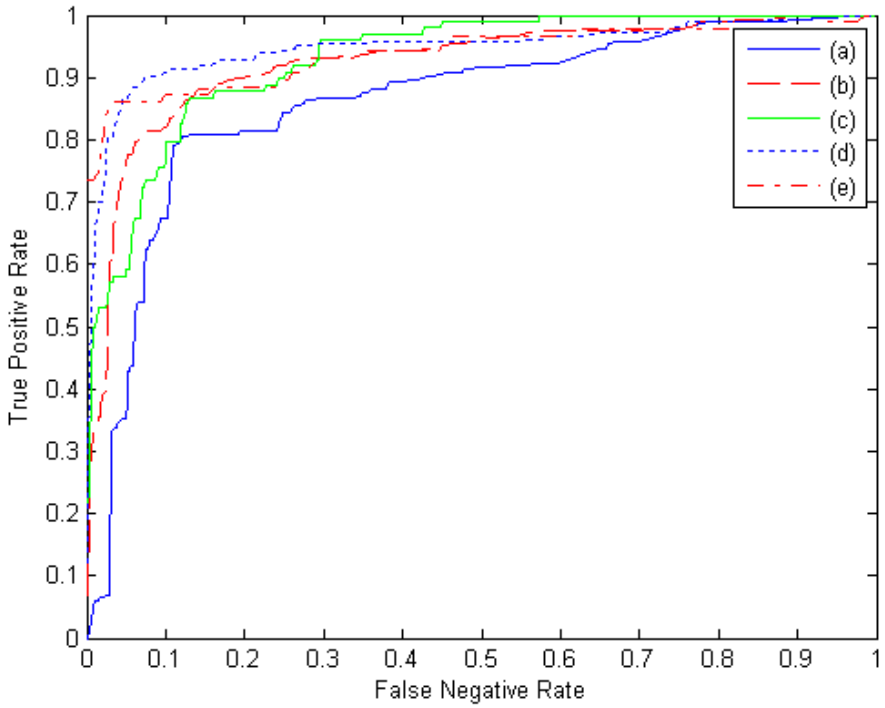


Figure 8. ROC curves for a representative test set for case 1 for the different classifiers with M=5, N=75 (except for Backpropagation ANN). (a) EANN GA algorithm. (b) EANN PSO algorithm. (c) EANNE Average algorithm. (d) EANNE LVQ algorithm. (e) Backpropagation ANN.

Table 2. Average area under the ROC curve test results and (precision/recall) for different algorithms and feature combinations. For the EANN and EANNE algorithms M refers to the population size, and N is the maximum number of generations.

Case 1		M=5,N=75	M=10,N=75	M=10,N=100	M=15,N=100
EANN	GA	0.862 (0.900/ 0.652)	0.867 (0.900/ 0.667)	0.865 (0.900/ 0.666)	0.884 (0.904/0.635)
	PSO	0.927 (0.900/0.835)	0.964 (0.900/ 0.922)	0.960 (0.905/ 0.942)	0.965 (0.906/0.941)
EANNE	Average	0.930 (0.900/0.783)	0.958 (0.900/0.865)	0.975 (0.906/0.972)	0.979 (0.900/0.970)
	LVQ	0.944 (0.901/ 0.838)	0.965 (0.903/0.890)	0.977 (0.906/0.972)	0.980 (0.906/0.973)
Backpropagation ANN		0.914 (0.900/0.795)			
Case 2					
EANN	GA	0.892 (0.900/0.550)	0.901 (0.900/0.540)	0.904 (0.900 0.686)	0.904 (0.900/0.686)
	PSO	0.876 (0.900/0.526)	0.917 (0.905/ 0.637)	0.924 (0.909/0.609)	0.914 (0.900/0.625)
EANNE	Average	0.884 (0.904/0.635)	0.884 (0.904/0.635)	0.893 (0.900/0.550)	0.902 (0.903/0.550)
	LVQ	0.884 (0.907/0.620)	0.883 (0.903/0.620)	0.896 (0.900/0.551)	0.905 (0.900/0.554)
Backpropagation ANN		0.905 (0.944/0.640)			
Case 3					
EANN	GA	0.792 (0.900/0.503)	0.855 (0.900/0.652)	0.830 (0.913/0.533)	0.821 (0.900/0.553)
	PSO	0.892 (0.900/0.782)	0.921 (0.901/0.820)	0.944 (0.900/0.864)	0.947 (0.905/0.850)
EANNE	Average	0.924 (0.902/0.861)	0.935 (0.900/0.860)	0.947 (0.900/0.855)	0.950 (0.900/0.861)
	LVQ	0.923 (0.902/0.861)	0.940 (0.900/0.861)	0.952 (0.900/0.863)	0.958 (0.900/0.865)
Backpropagation ANN		0.895 (0.944/0.681)			
Case 4					
EANN	GA	0.837 (0.913/0.540)	0.844 (0.923/0.553)	0.884 (0.904/0.635)	0.912 (0.900/0.624)
	PSO	0.954 (0.901/0.838)	0.961 (0.902/0.838)	0.974 (0.906/0.956)	0.978 (0.906/0.958)

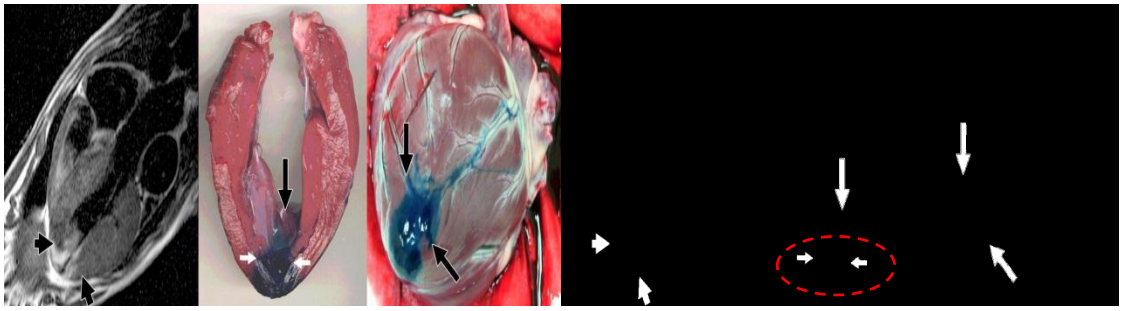
Table 2. Average area under the ROC curve test results and (precision/recall) for different algorithms and feature combinations. For the EANN and EANNE algorithms M refers to the population size, and N is the maximum number of generations (con't).

EANNE	Average	0.967 (0.900/0.950)	0.974 (0.900/0.960)	0.977 (0.906/0.958)	0.977 (0.906/0.958)
	LVQ	0.970 (0.900/0.958)	0.978 (0.906/0.958)	0.982 (0.903/0.892)	**0.988 (0.928/0.973)**
Backpropagation ANN		0.922 (0.917/0.830)			
Case 5					
EANN	GA	0.843 (0.923/0.553)	0.832 (0.932/0.516)	0.827 (0.932/0.514)	0.859 (0.900/0.655)
	PSO	0.942 (0.905/0.958)	0.974 (0.906/0.962)	0.963 (0.906/0.954)	0.985 (0.913/0.970)
EANNE	Average	0.956 (0.900/0.863)	0.976 (0.906/0.958)	0.984 (0.913/0.965)	0.980 (0.903/0.960)
	LVQ	0.956 (0.900/0.863)	0.977 (0.906/0.958)	0.985 (0.913/0.970)	0.986 (0.913/0.970)
Backpropagation ANN		0.909 (0.917/0.731)			
Case 6					
EANN	GA	0.828 (0.932/0.514)	0.831 (0.913/0.533)	0.896 (0.900/0.550)	0.873 (0.912/0.666)
	PSO	0.861 (0.900/ 0.667)	0.948 (0.906/0.958)	0.951 (0.905/0.940)	0.956 (0.900/0.963)
EANNE	Average	0.965 (0.906/0.954)	0.974 (0.906/0.958)	0.956 (0.900/0.863)	0.956 (0.900/0.963)
	LVQ	0.968 (0.906/0.955)	0.975 (0.906/0.958)	0.962 (0.906/0.954)	0.957 (0.900/0.963)
Backpropagation ANN		0.925 (0.917/ 0.865)			
Case 7					
EANN	GA	0.858 (0.900/0.667)	0.867 (0.906/0.667)	0.872 (0.912/0.667)	0.895 (0.900/0.550)
	PSO	0.924 (0.900/0.865)	0.961 (0.905/0.960)	0.976 (0.906/0.959)	0.985 (0.913/0.970)
EANNE	Average	0.978 (0.906/0.958)	0.975 (0.906/0.958)	0.974 (0.906/0.944)	0.976 (0.906/0.959)
	LVQ	0.975 (0.906/0.958)	0.979 (0.906/0.958)	0.983 (0.903/0.630)	0.983 (0.903/0.960)
Backpropagation ANN		0.958 (0.917/0.856)			

4. DISCUSSION

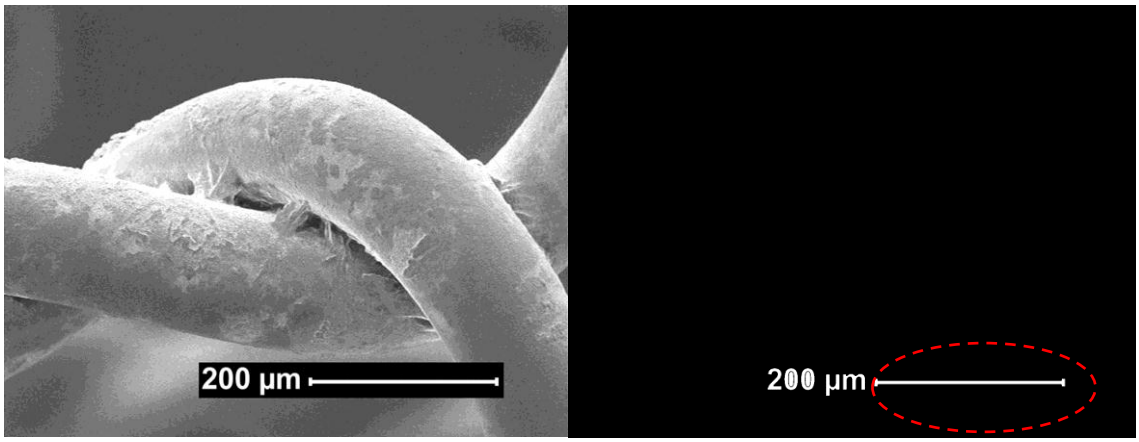
Inspecting Figure 8 and Table 2, several observations can be made. First, because the arrows in the experimental data set are uniform in grayscale, there were no cases of partially segmented arrow objects. Second, according to Figure 8, the classifier accuracy ranking from highest to lowest based on area under the ROC curve is EANNE LVQ algorithm, EANNE Average algorithm, EANN PSO algorithm, EANN GA algorithm, and backpropagation ANN. Third, the area under the ROC curve and precision/recall results are not directly related. The area under the ROC curve provides a measure of overall arrow discrimination capability over different classifier output thresholds. Precision and recall gives a measure of relevance to the objects labeled as arrows. Having a high area under the ROC curve does not always result in high precision/recall, as can be observed in Table 2. Having a high area under the ROC curve and high precision/recall demonstrates arrow objects can be successfully discriminated from non-arrow objects and that arrow objects are correctly found and not omitted in the selection/segmentation process within the medical images. The highest overall discrimination rates based on area under the ROC curve and precision/recall are 0.988 and 0.928/0.973, respectively, for the region property and shape features (case 4) using the EANNE with LVQ (winner-take-all) approach. Other feature combinations including all features (case 7) and the region property and correlation-based features (case 5) yielded similar results using the EANNE with LVQ approach. Fourth, the PSO algorithm consistently gives higher results, area under the ROC curve and precision/recall, than the GA approach for the EANN algorithm for the different feature combinations. Fifth, the EANN with PSO for weight updating and the EANNE methods consistently outperformed the standard backpropagation neural network benchmark approach for all feature combinations, highlighting the benefit of incorporating multiple neural networks in the training process. This is supported with the general observation that the area under ROC curve and precision/recall results are higher with more neural networks integrated into the training process, M = 10, 15 versus M = 5 for the EANN and EANNE classifiers. Sixth, for the EANNE approach, the LVQ winner-take-all method for integrating multiple neural networks gave consistently higher classification results than the averaging method for N=100 (last 2 columns of Table 2) for all feature combinations. Overall, the EANNE discrimination algorithms slightly outperform the EANN methods for the same input feature combinations. This experimental result highlights the robustness of the EANNE algorithms to the variations in the types of features explored for arrow discrimination as well as the size and shape variations of the arrows present in the experimental data set (Cheng, 2010).

The experimental results show that arrow discrimination can be performed at a high success rate using the arrow segmentation, feature extraction, and computational intelligence methods presented in this paper. The arrow-like object segmentation algorithm found all arrows with numerous false positive arrow-like objects, hence, the need for feature and discrimination analysis. The approach presented demonstrates the utility of integrating imaging and computational intelligence methods for object segmentation. The objects circled by the red in Figure 9 and Figure 10 provide some examples with incorrect classification. Since arrows are typically narrow and long, small arrows with large width may be incorrectly classified as noise, as shown in Figure 9. In addition, narrow and long noise objects may be mistaken for arrow objects, as shown in Figure 10.



(a) (b)

Figure 9. Arrow objects incorrect classified (Adapted from (Saeed, 2004)). (a) Original image. (b) Binary object mask image.



(a) (b)

Figure 10. Non-arrow objects incorrect classified (Adapted from (Schürmann, 2004)). (a) Original image. (b) Binary object mask image.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a process for integrating image and feature analysis and computational intelligence-based techniques for arrow discrimination in annotated medical images. The arrow discrimination results show the potential for merging imaging and computational intelligence methods for accurate arrow discrimination and segmentation based on object pruning, i.e. labeling objects of interest. Experimental results yielded area under the ROC curve and precision/recall as high as 0.988 and 0.928/0.973, respectively, using the EANNE approach with winner-take-all LVQ approach. Future work will involve integrating the detection of medical annotations into an overall approach for fusing data such as key words, modality of medical image and figure captions to improve the relevance of the search results for medical publication querying. Future work will involve determining and incorporating arrow orientation information to assist in the assessment process of this symbol in medical images.

ACKNOWLEDGEMENTS

This work was supported by the National Library of Medicine, NLM under contract number 276200800413P and the Intramural Research Program of the National Institutes of Health (NIH), NLM, and Lister Hill National Center for Biomedical Communications (LHNCBC).

We thank the organizers of ImageCLEFmed 2008 (Müller, 2010) for providing the images for our research, some of which have been reproduced in this article. See http://www.imageclef.org for details.

We appreciate the contributions and efforts of the anonymous IJHISI reviewers for this paper.

REFERENCES

1. Antani, S., Demner-Fushman, D., Li, J., Srinivasan, B.V., Thoma, G.R., "Exploring use of images in clinical articles for decision support in Evidence-Based Medicine". In Proceedings of SPIE-IS&T Electronic Imaging, San Jose, CA, 2008, 6815, pp. Q(1-10).

2. Bar-Ilan, J., "On the overlap, the precision and estimated recall of search engines: A case study of the query "Erdos" ". Scientometrics, 1998, 42(2), pp.207-208.

3. Caskey, C.I., Berg, W.A., Hamper, U.M., Sheth, S., Chang, B.W., Anderson, N.D., "Imaging spectrum of extracapsular silicone: correlation of US, MR imaging", mammographic, and histopathologic findings, Radiographics, 1999, 19, pp.S39-S51.

4. Cheng, B., Stanley, R.J., Antani, S., Thoma G.R., "A Novel Computational Intelligence-based Approach For Medical Image Artifacts Detection", In International Conference on Artificial Intelligence and Pattern Recognition, Orlando, FL, 2010, pp.113-20.

5. Cheng, B., Erdos, D., Stanley, R.J., Stoecker, W.V., Calcara, D., Gomez, D., "Automatic Detection of Basal Cell Carcinoma Using Telangiectasia Analysis in Dermoscopy Skin Lesion Images". Skin Research and Technology, 2011, 17(3), pp.278–287.

6. Demner-Fushman, D., Antani, S.K., Thoma, G.R., "Automatically Finding Images for Clinical Decision Support". In Workshop on Data Mining in Medicine. 7th IEEE Intl Conf on Data Mining, 2011, pp. 139-144.

7. Demner-Fushman, D., Antani, S.K., Simpson, M., Thoma, G.R., "Combining Medical Domain Ontological Knowledge and Low-level Image Features for Multimedia Indexing". In 2nd International "Language Resources for Content-Based Image Retrieval" Workshop (OntoImage 2008), part of 6th Language Resources and Evaluation Conference, 2008.

8. Demner-Fushman, D., Antani, S., Simpson, M., Thoma, G.R., "Annotation and Retrieval of Clinically Relevant Images", International Journal of Medical Informatics: Special Issue on Mining of Clinical and Biomedical Text and Data, 2009, 78(12), pp.e59-e674.

9. Deserno, T.M., Antani, S., Long, R., "Ontology of Gaps in Content-Based Image Retrieval", Journal of Digital Imaging, 2009, 22(2), pp.202-15.

10. Dov, D., Liu, W., "Automated CAD Conversion with the Machine Drawing Understanding System: Concepts, Algorithms, and Performance", IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans, 1999, 29(4), pp.411-416.

11. Dov, D., Liu, W., "The sparse pixel vectorization algorithm and its performance evaluation", IEEE Trans on Pattern Analysis and Machine Intelligence, 1999, 21, pp.202–215.

12. Fogarty, J., Baker, R.S., Hudson, S.E., "Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction", In Proceedings of Graphics interface, Canadian Human-Computer Communications Society, School of Computer

Science, University of Waterloo, Waterloo, Ontario Victoria, British Columbia, 2005, pp.129-136.

13. Fraser, J.D., Anderson, D.R., "Deep venous thrombosis: recent advances and optimal investigation with US". Radiology, 1999, 211, pp. 9-24.

14. Hanselman, D.C., Littlefield, B.L., "Mastering MATLAB 7", Upper Saddle River, NJ: Prentice Hall, 2004.

15. Holland, J.H., "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: University of Michigan Press, 1975.

16. Kennedy, J., Eberhart, R., "Particle swarm optimization", In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp.1942-1948.

17. Kohavi, R., "A study of cross-validation and bootstrap for accuracy estimation and model selection", In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, San Francisco, CA,1995, 2, pp.1137–1143.

18. Kohonen, T., "Self-Organizing Maps", 3rd edition. New York: Springer-Verlag, 1995.

19. Li, T., Tachibana, K., Kuroki, M., Mesahide, K., "Gene transfer with echo-enhanced contrast agents: comparison between albunex, optison, and levovist in mice—initial results", Radiology, 2003, 229, pp. 423-428.

20. Liu, Y., Yao, X., "Ensemble learning via negative correlation", Neural Networks, 1999, 12, pp. 1399–1404.

21. Müller, H., Clough, P., Deselaers, T., Caputo, B. (Eds)., "Experimental Evaluation in Visual Information Retrieval", The Information Retrieval Series, Berlin: Springer, 2010. 32.

22. Otsu, N., "A threshold selection method from gray-level histograms", IEEE Transactions on Systems, Man and Cybernetics, 1979, 9(1), 62-66.

23. Park, J., Rasheed. W., Beak, J., "Robot Navigation Using Camera by Identifying Arrow Signs", Grid and Pervasive Computing Workshops, GPC Workshops '08, The 3rd International Conference on Grid and Pervasive Computing, Kunming, 2008, pp.382-386.

24. Piper, J., Granum, E., "On fully automatic feature measurement for banded chromosome classification", Cytometry, 1989, 10, 242-255.

25. Saeed, M., Lee, R., Martin, A., Weber, O., Krombach, G.A., Schalla, S., Lee, M. Saloner, D., Higgins, C.B., "Transendocardial delivery of extracellular myocardial markers by using combination x-ray/MR fluoroscopic guidance: feasibility study in dogs", Radiology, 2004, 231, pp.689-696.

26. Serra, J., "Image Analysis and Mathematical Morphology", London: Academic Press, 1982, 1.

27. Schürmann, K., Lahann, J., Niggermann, P., Klosterhalfen, B., Meyer, J., Kulisch, A., Klee, D., Günther, R.W., Vorwerk, D., "Biologic response to polymer-coated stents: in vitro analysis and results in an Iliac artery sheep model", Radiology, 2004, 230, pp.151-162.

28. Stanley, R.J., Stoecker, W.V., & Moss, R.H., "A basis function feature-based approach for skin lesion discrimination in dermatology dermoscopy images", Skin Research and Technology, 2008, 14(4), pp.425-435.

29. Stanley, R.J., Gader, P., Ho, D., "Feature and decision level sensor fusion of electromagnetic induction and ground penetrating radar sensors for landmine detection with hand-held units", Information Fusion, 2002, 3(3), pp.215-223.

30. Yao, X., "A review of evolutionary artificial neural networks", International Journal of Intelligent Systems, 1993, 8(4), pp.539–567.

31. Yao, X., Islam, M.M., "Evolving artificial neural network ensembles", Computational Intelligence Magazine, 2008, 3(1), 31-42.

32. You, D., Apostolova, E., Antani, S.K., Demner-Fushman, D., Thoma, G.R., "Figure content analysis for improved biomedical article retrieval", In SPIE-IS&T Electronic Imaging, San Jose, CA, 2008, 7247, pp. v(1-10).

33. You, D., Antani, S.K., Demner-Fushman, D., Rahman, M.M., Govindaraju, V, Thoma, G.R., "Biomedical article retrieval using multimodal features and image annotations in region-based CBIR", In SPIE-IS&T Electronic Imaging, San Jose, CA, 2010, 7534, pp.v(1-12).

APPENDIX

A.1 EVOLVING ARTIFICIAL NEURAL NETWORKS

A.1.1 Evolving Artificial Neural Networks trained by Particle Swarm Optimization. Particle Swarm Optimization (PSO) (Kennedy, 1995) is the study of swarms of social organisms such as flock of birds, which each particle in the swarm moves toward its previous best location (*Pbest*) and global best location (*Gbest*) defined below at each time step. To train the connection weights in the ANNs, each candidate is a particle. *Pbest* is the particle of the *M* particles that gives the least RMSE between the current epoch of ANN's training and the previous epoch with ANN's training. *Gbest* is the particle among the *M* particles which generates the minimum RMSE. The velocity to update the particle is presented in Equation 6. The position vector of the particles is changed as shown in Equation 7. The same process is used for obtaining the next set of particles, which is repeated by *N* epochs.

The velocity of the particles is given as follows:

$$V_{md}(n+1) = wV_{md}(n) + c_1 rand_1\left(Pbest_{md} - X_{md}(n)\right) + c_2 rand_2\left(Pbest_{md} - X_{md}(n)\right) \quad (6)$$

The position vector of the particles is changed as follows:

$$X_{md}(n+1) = X_{md}(n) + V_{md}(n+1) \quad (7)$$

where *n* is the current iteration (time step) *(* $1 \le n \le N$ *)*, *m* is the current particle *(*$1 \le m \le M$*)*, *d* is the weight element (*1*≤*d*≤ number of weights), $V_{md}(n)$ is the particle's current velocity, $V_{md}(n+1)$ is the particle's new velocity, $X_{md}(n)$ is the particle's current position, $X_{md}(n+1)$ is the particle's new position, $rand_1$ and $rand_2$ are the random values selected from *0* to *1*, *w* is the inertia weight chosen as *0.7*, c_1 is the cognitive acceleration constant of *1.5*, and c_2 is the social acceleration constant of *1.5*.

A.1.2 Evolving Artificial Neural Networks trained by Genetic Algorithm. Genetic Algorithm (GA) (Holland, 1975) provides optimization by using selection, crossover, mutation and elitism operators. The implementation used in this research consists of generating M offspring, i.e. particles from a pool of M sets of initial weights comprising a parent pool.

The offspring are generated as follows: 1) randomly select two parents (sets of weights)

from the parent pool of M sets of initial weights; 2) initialize offspring weights as the first parent weights and apply a randomly generated binary mask with the size of the weights matrix for the crossover process in order to recombine selected parents to get offspring. For this randomly generated binary mask, if a random value is selected which is less than 0.5, then the bit of binary mask is 0, otherwise, it is 1; 3) apply a mutation process for weight updating. The mutation process consists of selecting a weight and then adding a random value to it. A weight is chosen if a random value is less than 0.5, then another random value between -1 to +1 is added to the weight value.

MLP training is performed using the parent pool of weights and the offspring pool of weights based on the ANN architecture above. The next parent pool is chosen based on whether the parent is used for initialization or its offspring minimizes the RMSE error. The same process is used for obtaining the next set of offspring, which is repeated by N epochs. From the final parent pool, the parent which minimizes the RMSE error over the training feature vectors is selected for the final ANN weights for the test vectors.

A.2 EVOLVING ARTIFICIAL NEURAL NETWORK ENSEMBLES

A.2.1 Training the Networks. A learning paradigm named negative correlation learning (NCL) is used for training neural network ensembles. The idea of negative correlation learning is to introduce a correlation penalty term into the error function of each individual network so that the mutual information among the networks in the ensemble can be minimized (Liu, 1999).

The steps of genetic algorithm neural network ensembles for training are given as follows. First, generate an initial population of *M* ANNs, and set the iteration number *n* to be 1, the random initial weights are distributed uniformly inside a small range. Second, train each ANN in the initial population on the training set for a certain number of epochs using negative correlation learning. Third, calculate the fitness of *M* ANNs in the population. Fourth, create *M* offspring ANNs by using selection, crossover, and mutation. Fifth, replace the worst *M* ANNs in the current population with *M* offspring ANNs, and train the whole population using negative correlation learning for another epoch. Sixth, stop the evolution process if the maximum number of iterations (*N*) has been reached. Otherwise, $n = n + 1$ and go to step 2.

A.2.2 Final output decision. Once a population of networks has been trained, a mechanism is needed to decide the final output based on the outputs from individual networks. Different methods are considered such as averaging, majority voting, and winner-taking-all. Since the outputs of the ANNs are floating-point numbers, we explored averaging and winner-taking-all for combining/aggregating the outputs. For averaging, the output (y_{avg}) is simply expressed as follows:

$$y_{avg} = \frac{1}{M} \sum_{m=1}^{M} F_m \tag{8}$$

where, *M* is the number of the individual ANNs in the ensemble.

For winner-taking-all approach, the output of the network with the strongest activation is chosen. A learning vector quantization network (LVQ) is trained after training the neural network ensembles (Kohonen, 1995). There are two layers in LVQ-competitive layer and output layer. The net output of the first layer of the LVQ is given by W, expressed as:

$$L_1 = argmin_m ||x - w_{1m}|| \tag{9}$$

where x is the same input vector as the input to the ensemble of ANNs, $m=1,2,\ldots,M$ and w_{1m} is the weight of the m^{th} neuron in the first layer.

The network output of the second layer of the LVQ is given by:

$$L_2 = w_2 L_1 \tag{10}$$

where w_2 is the weight in the second layer. The second layer of the LVQ network is used to combine subclasses into a single class. The columns of w_2 represent the subclasses (*M*) and the rows of the matrix represent the classes (*C*). w_2 has a single *1* in each column, with others elements set to zero. The value *1* in each row indicates which class (the row number) the appropriate subclass (the column number) should be combined into. We set *C* equal to *M*.
The training target (*T*) is given as:

$$T = \begin{cases} argmin(F_m), \ if \ class \ 0 \\ argmax(F_m), \ if \ class \ 1 \end{cases} \tag{11}$$

We adjust the synaptic weight vectors of all neurons by using the update formula on the nth training epoch:

$$w_{1m}(n+1) = w_{1m}(n) + \eta(n)(x - w_{1m}(n)) \quad (12)$$

where, $\eta(n)$ is the learning rate which is set to be *0.2* if L_2 is the same as the *T*, otherwise, it is *-0.2*.

Therefore, we have a trained LVQ network with the same input as EANNs and the winner neural network number as the output (*1, 2,…, or M*) indicated by L_2.